

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L4	104	XML with print\$3	USPAT	OR	OFF	2006/02/17 14:25
L5	9	XML with print\$3 with driver	USPAT	OR	OFF	2006/02/17 14:31
L6	13	XML same print\$3 with driver	USPAT	OR	OFF	2006/02/17 14:31
L7	176	"6012098" "6154738"	USPAT	OR	OFF	2006/02/17 15:24
S1	3	"6,763,343"	USPAT	OR	OFF	2006/02/13 14:32
S2	458	xsl	USPAT	OR	OFF	2006/02/09 17:31
S3	57	xsl with code\$1	USPAT	OR	OFF	2006/02/09 17:31
S4	82	xml and xsl and xslt	USPAT	OR	OFF	2006/02/13 15:29
S5	21	"6918084" "6487566" "6832351" "6763343" "6631375" "6507410"	USPAT	OR	OFF	2006/02/13 15:29
S6	6	("6918084" "6487566" "6832351" "6763343" "6631375" "6507410"). pn.	USPAT	OR	OFF	2006/02/17 14:25
S7	1094	print\$3 with ML	USPAT	OR	OFF	2006/02/13 15:33
S8	102	print\$3 with xml	USPAT	OR	OFF	2006/02/13 15:34
S9	6	("5860068" "6154378" "6226675" "6289501" "6449620" "6457883").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/02/13 15:37
S10	5	S9 and xml	US-PGPUB; USPAT; USOCR	OR	OFF	2006/02/13 15:37
S11	56	xsl with code	USPAT	OR	OFF	2006/02/14 13:47
S12	81	xsl with (code program)	USPAT	OR	OFF	2006/02/14 13:49
S13	58	xsl with (code program) and print\$3	USPAT	OR	OFF	2006/02/17 13:42

Save Yo
[Why You Need a](#)
[How You Can C](#)

XML Index

XML GUIDE in PDF Form

1. [Introduction](#)
2. [XML Uses](#)
3. [XML Categories](#)

In this XSL example, I create two lists in XML and use the XSL style sheet to set the styl

Structure

1. [Elements](#)
2. [Document Types](#)
3. [Document Structure](#)
4. [Document Type Definition \(DTD\)](#)
5. [An Example](#)
6. [Element Attributes](#)
7. [Process Instructions](#)
8. [Extensible linking Language \(XLL\)](#)

The XML File

```
<?xml version="1.0"?>
<!DOCTYPE LANGLIST SYSTEM "langlist.dtd">
<?xml-stylesheet type="text/xsl" href="xmlstyle.xsl"?>
<LANGLIST>
  <TITLE>List of Items Important to Markup Languages</TITLE>
  <TITLE1>Languages</TITLE1>
    <LIST1>
      <LANGUAGES>SGML</LANGUAGES>
      <LANGUAGES>XML</LANGUAGES>
      <LANGUAGES>HTML</LANGUAGES>
    </LIST1>
  <TITLE2>Other Support</TITLE2>
    <LIST2>
      <OTHER>DTD</OTHER>
      <OTHER>DSSSL</OTHER>
      <OTHER>Style Sheets</OTHER>
    </LIST2>
</LANGLIST>
```

The DTD File

```
<!ELEMENT LANGLIST ANY>
<!ENTITY % Shape "(rect|circle|poly|default)">
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT TITLE1 (#PCDATA)>
<!ELEMENT TITLE2 (#PCDATA)>
<!ELEMENT LIST1 (LANGUAGES)+>
<!ELEMENT LIST2 (OTHER)+>
<!ELEMENT LANGUAGES (#PCDATA)>
<!ATTLIST LANGUAGES
  type %Shape; #IMPLIED>
<!ELEMENT OTHER (#PCDATA)>
<!ATTLIST OTHER
  type (disc|square|circle) #IMPLIED>
```

Use

1. [Binding to HTML](#)

Other

XSL

1. [XSL Elements](#)
2. [XSL Example](#)

DOM

3. [DOM](#) This DTD file employs an ENTITY, just for demonstration purposes. An entity, in this cas

Implementations the LANGUAGES and OTHER elements, although these attributes are not used in this example.

- [OSD](#)

The XML Style File

Appendices

1. [Terms](#)
2. [Credits](#)

XML

Web Tech

Independent

Home

```
<?xml version="1.0"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```

    <xsl:template match="/">
        <xsl:apply-templates select="LANGLIST/TITLE" />
        <xsl:apply-templates select="LANGLIST/TITLE1" />
        <xsl:apply-templates select="LANGLIST/LIST1" />
        <xsl:apply-templates select="LANGLIST/TITLE2" />
        <xsl:apply-templates select="LANGLIST/LIST2" />
    </xsl:template>

    <xsl:template match="TITLE">
        <SPAN STYLE="display: 'block'; font-family: 'arial'; col.
        <xsl:value-of />
        </SPAN>
        <BR/>
    </xsl:template>

    <xsl:template match="TITLE1">
        <SPAN STYLE="display: 'block'; font-family: 'arial'; col.
        <xsl:value-of />
        </SPAN>
        <BR/>
    </xsl:template>

    <xsl:template match="LIST1">
        <UL style="display: 'list-item'; list-style-image: url('
        <xsl:for-each select="LANGUAGES">
            <LI style="display: 'list-item'; list-style-type
            <xsl:value-of />
            </LI>
        </xsl:for-each>
        </UL>
    </xsl:template>

    <xsl:template match="TITLE2">
        <SPAN STYLE="display: 'block'; font-family: 'arial'; col.
        <xsl:value-of />
        </SPAN>
        <BR/>
    </xsl:template>

    <xsl:template match="LIST2">
        <UL style="display: 'list-item'; list-style-image: url('
        <xsl:for-each select="OTHER">
            <LI style="display: 'list-item'; list-style-type
            <xsl:value-of "." />
            </LI>
        </xsl:for-each>
        </UL>
    </xsl:template>

</xsl:stylesheet>
```

In the section above between the `<xsl:template match="/">` tag and `</xsl:template>` tag, the content is processed in order to display the value of that object.

[Click Here to see the LONGLIST.XML file \(USE your browser's BACK button to return\).](#)